Candidate Name : Jitendra Rajendra Jivrak MCA II

PART 1: Problem Statement

Problem Statement : Alert Triage Workflow: Show how a security engineer might investigate and resolve a cloud security alert.

Cloud security platforms generate hundreds of alerts daily across Kubernetes clusters, cloud workloads, and containers. Security engineers often struggle with alert fatigue due to duplicate, low-context, and unprioritized alerts.

This leads to:

- Delayed response to critical threats
- MTTR - Increased Mean Time to Resolve
- Inefficient investigation workflow
- Higher risk of production impact

There is a need for a simple alert triage workflow that helps security engineers prioritize, investigate, and resolve security alerts efficiently.

PART 2: User Persona

Primary Persona : Security Engineer

Environment:

- Manages multi-cloud setup like AWS, GCP, Azure
- Handles Kubernetes security alerts
- Uses multiple dashboards daily

Goals:

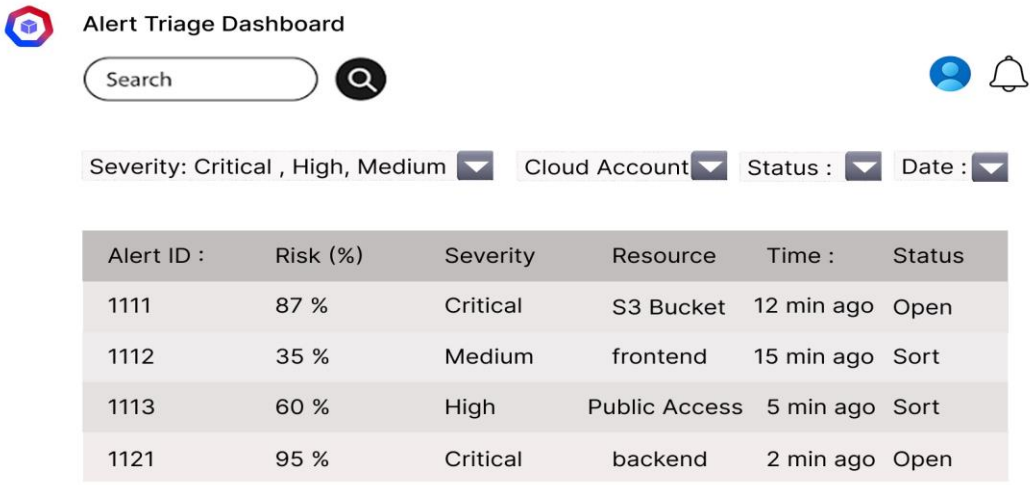
- Identify real threats quickly
- Reduce complexity from duplicate alerts
- Assign and track alert ownership
- Resolve issues before Service level Agreements SLA breach

Pain Points:

- Too many alerts with no context
- No prioritization beyond severity
- Manual investigation steps
- No workflow tracking

PART 3: My 3 Figma Screens

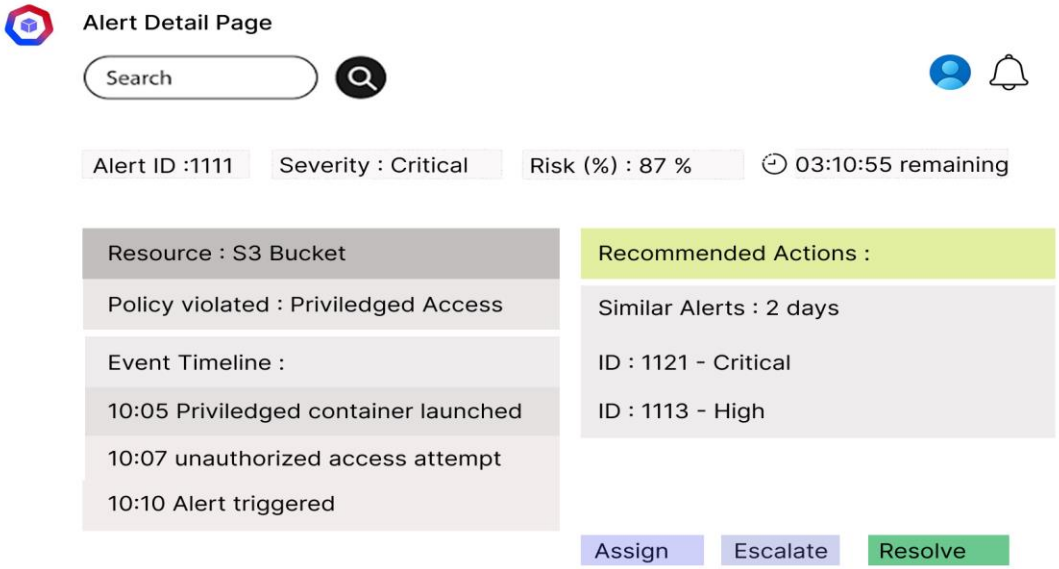
Screen 1: Alerts Dashboard



Annotations :

- Risk Score improves prioritization beyond severity
- Filters reduce investigation time
- Grouping similar alerts reduces noise

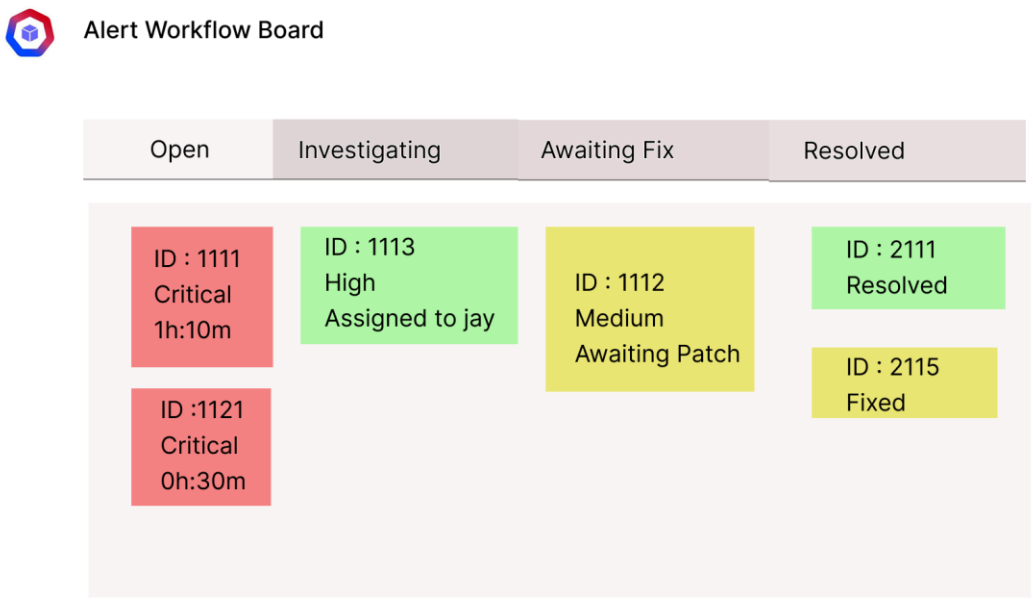
Screen 2: Alert Detail View



Annotations:

- Contextual information centralized
- SLA timer ensures accountability
- Recommended remediation reduces decision delay

Screen 3: Workflow Board



Annotations:

- Kanban view improves visibility
- Ownership tracking reduces backlog
- Status tracking prevents SLA breaches

PART 4 : Design Rationale

a structured workflow approach is :

Identify → Investigate → Resolve

1. The dashboard focuses on quick scanning and prioritization.
2. The detail view centralizes investigation context to reduce tool switching.
3. The workflow board introduces transparency and ownership tracking.

The overall design minimizes load and supports rapid decision-making in high-pressure security environments.

PART 5 : Feature Prioritization

Prioritization Approach: RICE Framework

MVP Features:

- Centralized alert dashboard
- Risk scoring mechanism
- Alert detail investigation page
- Assignment and status tracking

Phase 2 Enhancements:

- AI-based alert clustering
- Predictive risk scoring
- Integration with CI/CD pipelines

PART 6 : Success Metrics

The effectiveness of this solution can be measured by:

- 30% reduction in MTTR
- 40% reduction in duplicate alerts
- Increase in alerts resolved within SLA
- Reduction in alert backlog
- Improved user satisfaction score

PART 7 : Development Discussion Points

To implement this solution, the following technical considerations are required:

- Cloud alert ingestion pipeline
- Risk scoring algorithm (severity + asset criticality)
- Role-based access control (RBAC)
- SLA tracking microservice
- Alert clustering logic
- Audit logging system